ACE
Engineering College
An Autonomous Institution
All the courses are Accredited by NBA and NAAC with A Grade, Affiliated to JNTUH
Ankushapur, Ghatkesar ,Medchal,Hyderabad-501301

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE STRUCTURE – R22

III YEAR II SEMESTER

S. No.	Course Code	Course Title	L	T	P	Credits
1	IT601PC	Automata Theory and Compiler Design	3	0	0	3
2	IT602PC	Algorithms Design and Analysis	3	0	0	3
3	IT603PC	Embedded Systems	3	0	0	3
4		Professional Elective –III	3	0	0	3
5		Open Elective-I	3	0	0	3
6	IT604PC	Compiler Design Lab	0	0	2	1
7	IT605PC	Embedded Systems Lab	0	0	2	1
8		Professional Elective-III Lab	0	0	2	1
9	IT606PC	Industrial Oriented Mini Project/ Internship/ Skill Development Course (Big data-Spark)	0	0	4	2
10	*MC609	Environmental Science	3	0	0	0
		Total	**18**	**0**	**10**	**20**

Environmental Science in III Yr II Sem Should be Registered by Lateral Entry Students Only.

#Skill Course - 1 credit with 2 Practical Hours

Professional Elective - Ill

IT631PE	Full Stack Development
IT632PE	Data Mining
IT633PE	Scripting Languages
IT634PE	Mobile Application Development
IT635PE	Software Testing Methodologies

# **Courses in PE - Ill and PE -** Ill **Lab must be in 1-1 correspondence.**

Open Elective -1:

1. IT611OE: Java Programming
2. IT612OE: Object Oriented Programming using C++

IT601PC: AUTOMATA THEORY AND COMPILER DESIGN

B.Tech. III Year II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Objectives

- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- To understand deterministic and non-deterministic machines and the differences betweendecidability and undecidability.
- Introduce the major concepts of language translation and compiler design and impart theknowledge of practical skills necessary for constructing a compiler.
- Topics include phases of compiler, parsing, syntax directed translation, type checking use of symbol tables, intermediate code generation

Course Outcomes

- Able to employ finite state machines for modeling and solving computing problems.
- Able to design context free grammars for formal languages.
- Able to distinguish between decidability and undecidability.
- Demonstrate the knowledge of patterns, tokens & regular expressions for lexical analysis.
- Acquire skills in using lex tool and design LR parsers.
- Able to generate intermediate-code representation .

UNIT - I

Introduction to Finite Automata: Structural Representations, Automata and Complexity, the Central Concepts of Automata Theory – Alphabets, Strings, Languages, Problems.

Nondeterministic Finite Automata: Formal Definition, an application, Text Search, Finite Automatawith Epsilon-Transitions.

Deterministic Finite Automata: Definition of DFA, How A DFA Process Strings, The language ofDFA, Conversion of NFA with €-transitions to NFA without €-transitions. Conversion of NFA to DFA.

UNIT - II

Regular Expressions: Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Conversion of Finite Automata to Regular Expressions.

Pumping Lemma for Regular Languages:

Statement of the pumping lemma, Applications of the Pumping Lemma.

Context-Free Grammars: Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Parse Trees, Ambiguity in Grammars and Languages.

UNIT - III

Push Down Automata: Definition of the Pushdown Automaton, the Languages of a PDA, Equivalenceof PDA's and CFG's, Acceptance by final state.

Turing Machines:

Introduction to Turing Machine, Formal Description, Instantaneous description, The language of a Turing machine.

Undecidability:

Undecidability, A Language that is Not Recursively Enumerable, An Undecidable Problem That isRE, Undecidable Problems about Turing Machines.

UNIT - IV

Introduction: The structure of a compiler.

Lexical Analysis: The Role of the Lexical Analyzer, Input Buffering, Recognition of Tokens, TheLexical- Analyzer Generator Lex.

Syntax Analysis: Introduction, Context-Free Grammars, Writing a Grammar,

Top Down Parsing: Introduction, Recursive Descent, Predictive parser, LL(1) Parser.

Bottom- Up Parsing, Introduction to LR Parsing: Simple LR, More Powerful LR Parsers.

UNIT - V

Syntax-Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD's, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD's.
Intermediate-Code Generation: Variants of Syntax Trees, Three-Address Code
Run-Time Environments: Stack Allocation of Space, Access to Nonlocal Data on the Stack, HeapManagement

TEXT BOOKS:

1. Introduction to Automata Theory, Languages, and Computation, 3rd Edition, John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Pearson Education.
2. Theory of Computer Science- Automata languages and computation, Mishra and Chandrashekaran, 2nd Edition, PHI.

REFERENCE BOOKS:

1. Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, 2nd Edition, Pearson.
2. Introduction to Formal languages Automata Theory and Computation, Kamala Krithivasan, Rama R, Pearson.
3. Introduction to Languages and The Theory of Computation, John C Martin, TMH.
4. lex & yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
5. Compiler Construction, Kenneth C. Louden, Thomson. Course Technology.

IT602PC: ALGORITHMS DESIGN AND ANALYSIS

B.Tech. III Year II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Prerequisites: Programming for problem solving and Data Structures

Course Objectives:

- Introduces the notations for analysis of the performance of algorithms.
- Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which eachtechnique is appropriate.
- Describes how to evaluate and compare different algorithms using worst, average and best-case analysis.
- Explains the difference between tractable and intractable problems, and introduces theproblems that are P, NP and NP complete.

Course Outcomes:

- Analyze the performance of algorithms.
- Choose appropriate data structures and algorithm design methods for a specified application
- Understand the choice of data structures and the algorithm design methods
- Explain important algorithmic design paradigms and apply when an algorithmic design situation calls for it.
- Understand, solve and analyze various graph applications.
- Prove the hardness of NP-Hard problems using simple reductions.

UNIT - I

Introduction: Algorithm, Performance Analysis-Space complexity, Time complexity, AsymptoticNotations- Big oh notation, Omega notation, Theta notation and Little oh notation.
Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Strassen'smatrix multiplication.

UNIT - II

Disjoint Sets: Disjoint set operations, union and find algorithms, Priority Queue- Heaps, Heapsort **Backtracking:** General method, applications, n-queen's problem, sum of subsets problem, graphColoring, Hamiltonian cycles.

UNIT - III

Dynamic Programming: General method, applications- Optimal binary search tree, 0/1 knapsackproblem, All pairs shortest path problem, Traveling sales person problem, Reliability design.

UNIT - IV

Greedy method: General method, applications- Job sequencing with deadlines, knapsack problem,Minimum cost spanning trees, Single source shortest path problem.
Basic Traversal and Search Techniques: Techniques for Binary Trees, Techniques for Graphs,Connected components, Biconnected components.

UNIT - V

Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsackproblem - LC Branch and Bound solution, FIFO Branch and Bound solution.
NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP - Hard andNP-Complete classes, Cook's theorem.

TEXT BOOK:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharan,University Press.

REFERENCE BOOKS:

1. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson education.
2. Introduction to Algorithms, second edition, T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C.Stein, PHI Pvt. Ltd./ Pearson Education.
3. Algorithm Design: Foundations, Analysis and Internet Examples, M.T. Goodrich andR.Tamassia, John Wiley and sons.

IT603PC: EMBEDDED SYSTEMS

B.Tech. III Year II Sem. **L T P C**
3 0 0 3

Pre-requisites:

1. A course on “Digital Logic Design and Microprocessors”.
2. A course on “Computer Organization and Architecture”.

Course Objectives:

- To provide an overview of principles of Embedded System
- To provide a clear understanding of role of firmware, operating systems in correlation withhardware systems.

Course Outcomes:

- Expected to understand the selection procedure of processors in the embedded domain.
- Design procedure of embedded firm ware.
- Expected to visualize the role of realtime operating systems in embedded systems.
- Expected to evaluate the correlation between task synchronization and latency issues
- Expected to understand the protocols of different communication interfaces/devices.
- Expected to understand the selection of the memory in the designing of the Embedded System.

UNIT - I

Introduction to Embedded Systems: Processor embedded into a system, Embedded Hardware units and devices in a system, Embedded software in a system, Design process of an embedded system, classification of embedded systems, characteristics and quality attributes of an embedded systems

UNIT - II

Introduction to processor/microcontroller architecture, Real world interfacing, processor and memory organization, memory types, memory maps and addresses, interrupt sources and interrupt service mechanism.

UNIT - III

On board Communication Basics: serial; communication devices, Parallel devices, Wireless devices, Real time clock, Serial bus communication Protocols-I2C, SPI; Parallel buss communication-ISA, PCI.

UNIT - IV

Embedded Firmware Development: Overview of programming concepts - in assembly language and in high level language ‘C’, C Program elements- Heads, Source files, Processor Directives, Macros, Functions, Data types and Data Structures

UNIT - V

OS Based Embedded Systems: OS services - Process/Task Management, Memory Management, I/O subsystem manager, Inter Process/Task communications - Tasks, Task states, Shared data, Signals, Message Queues, Mailbox, Pipes and concepts of Semaphores.

TEXT BOOK:

1. Embedded Systems, Raj Kamal, 2nd edition, Tata Mc Graw Hill
2. Shibu K V, “Introduction to Embedded Systems”, Second Edition, Mc Graw Hill

REFERENCE BOOKS:

1. Rajkamal, Embedded Systems Architecture, Programming and Design, Tata McGraw-Hill
2. Frank Vahid and Tony Givargis, “Embedded Systems Design” - A Unified Hardware/SoftwareIntroduction, John Wiley
3. Lyla, “Embedded Systems” –Pearson
4. David E. Simon, An Embedded Software Primer, Pearson Education Asia, First Indian Reprint2000.

IT631PE: FULL STACK DEVELOPMENT (Professional Elective – III)

B.Tech. III Year II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Pre-Requisites:

1. Object Oriented Programming
2. Web Technologies

Course Objectives:

- Students will become familiar to implement fast, efficient, interactive and scalable webapplications using run time environment provided by the full stack components.

Course Outcomes:

1. Understand Full stack components for developing web application.
2. Apply packages of NodeJS to work with Data, Files, Http Requests and Responses.
3. Use MongoDB data base for storing and processing huge data and connects with NodeJS application.
4. Understanding the importance of building a robust backend in a full-stack development environment.
5. Design faster and effective single page applications using Express and Angular.
6. Create interactive user interfaces with react components

UNIT-I

Introduction to Full Stack Development:

Understanding the Basic Web Development Framework- User, Browser, Webserver, Backend Services, Full Stack Components - Node.js, MongoDB, Express, React, Angular. Java Script Fundamentals, NodeJS- Understanding Node.js, Installing Node.js, Working with Node Packages, creating a Node.js Application, Understanding the Node.js Event Model, Adding Work to the Event Queue, Implementing Callbacks

UNIT-II

Node.js:

Working with JSON, Using the Buffer Module to Buffer Data, Using the Stream Module to Stream Data, Accessing the File System from Node.js- Opening, Closing, Writing, Reading Files and other File System Tasks. Implementing HTTP Services in Node.js- Processing URLs, Processing Query Strings and Form Parameters, Understanding Request, Response, and Server Objects, Implementing HTTP Clients and Servers in Node.js, Implementing HTTPS Servers and Clients. Using Additional Node.js Modules- Using the os Module, Using the util Module, Using the dns Module, Using the crypto Module.

UNIT-III

MongoDB:

Need of NoSQL, Understanding MongoDB, MongoDB Data Types, Planning Your Data Model, Building the MongoDB Environment, Administering User Accounts, Configuring Access Control, Administering Databases, Managing Collections, Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, Understanding the Objects Used in the MongoDB Node.js Driver, Accessing and Manipulating Databases, Accessing and Manipulating Collections

UNIT-IV

Express and Angular:

Getting Started with Express, Configuring Routes, Using Requests Objects, Using Response Objects. Angular: importance of Angular, Understanding Angular, creating a

Basic Angular Application, Angular Components, Expressions, Data Binding, Built-in Directives, Custom Directives, Implementing AngularServices in Web Applications.

UNIT-V

React:

Need of React, Simple React Structure, The Virtual DOM, React Components, Introducing React Components, Creating Components in React, Data and Data Flow in React, Rendering and Life Cycle Methods in React, Working with forms in React, integrating third party libraries, Routing in React.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular WebDevelopment, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas, React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo,Express, React, and Node, 2nd Edition, Apress, 2019.
2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday SkillsExpected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.
3. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications UsingReact and Redux, 2nd edition, Addison-Wesley Professional, 2018.

IT632PE: DATA MINING (Professional Elective – III)

B.Tech. III Year II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Pre-Requisites:

1. Database Management System
2. Probability and Statistics

Course Objectives:

- Students will become acquainted with both the strengths and limitations of various data miningtechniques like Association, Classification, Cluster and Outlier analysis.

Course Outcomes:

- Understand the need of data mining and pre-processing techniques.
- Perform market basket analysis using association rule mining.
- Understand the pattern mining techniques in various association levels.
- Utilize classification techniques for analysis and interpretation of data.
- Identify appropriate clustering and outlier detection techniques to handle complex data.
- Understand the mining of data from web, text and time series data.

UNIT - I

Introduction to Data Mining:

What is Data mining? Kinds of Data, Knowledge Discovery process, Data Mining Functionalities, Kinds of Patterns, Major Issues in Data Mining. Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity.

Data Pre-processing: Major Tasks in Data Pre-processing, Data Cleaning, Data Integration, Data Reduction, Data Transformationand Data Discretization.

UNIT - II

Association Analysis: Basic Concepts, Market Basket Analysis, Apriori Algorithm, FP-growth, From Association Analysis to Correlation Analysis, Pattern Mining in Multilevel Associations and Multidimensional Associations.

UNIT - III

Classification: Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, Metrics for Evaluating Classifier Performance, Ensemble Methods, Multilayer Feed- Forward Neural Network, Support Vector Machines, k-Nearest-Neighbor Classifiers.

UNIT - IV

Cluster Analysis: Requirements for Cluster Analysis, Overview of Basic Clustering Methods, Partitioning Methods-k-Means, k-Medoids, Hierarchical Methods-AGENES, DIANA, BIRCH, Density- Based Method-DBSCAN.

Outlier Analysis: Types of Outliers, Challenges of Outlier Detection, and Overview of Outlier Detection Methods.

UNIT - V

Advanced Concepts: Web Mining- Web Content Mining, Web Structure Mining, Web Usage Mining, Spatial Mining- Spatial Data Overview, Spatial Data Mining Primitives, Spatial Rules, Spatial Classification Algorithm, Spatial Clustering Algorithms, Temporal Mining- Modeling Temporal Events, Time Series, Pattern Detection, Sequences, Temporal Association Rules.

TEXT BOOKS:

1. Jiawei Han, Micheline Kamber, Jian Pei., Data Mining: Concepts and Techniques, 3rd Edition,Morgan Kaufmann/Elsevier, 2012.
2. Margaret H Dunham, Data Mining Introductory and Advanced Topics, 2nd Edition, PearsonEducation, India, 2006.

REFERENCE BOOKS:

1. Data Mining Techniques, Arun K Pujari, 3rd Edition, Universities Press.
2. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne and Vipin Kumar, Introduction to DataMining, 2nd Edition, Pearson Education India, 2021.
3. Amitesh Sinha, Data Warehousing, Thomson Learning, India, 2007.

IT633PE: SCRIPTING LANGUAGES (Professional Elective – III)

B.Tech. III Year II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Prerequisites:

1. A course on "Computer Programming and Data Structures".
2. A course on "Object Oriented Programming Concepts".

Course Objectives:

- This course introduces the script programming paradigm
- Introduces scripting languages such as Perl, Ruby and TCL.
- Learning TCL

Course Outcomes:

- Comprehend the differences between typical scripting languages and typical system andapplication programming languages.
- Gain knowledge of the strengths and weakness of Perl, TCL and Ruby; and select anappropriate language for solving a given problem.
- Acquire programming skills in scripting language.
- Understand fundamental concepts of Tk and visual toolkits.
- Understand memory allocation, the Ruby Type System, and embedding a Ruby interpreter in other languages.
- Implement event handling, create visual interfaces, and address security concerns in Perl-Tk applications.

UNIT - I

Introduction: Ruby, Rails, The structure and Execution of Ruby Programs, Package Management with RUBYGEMS, Ruby and web: Writing CGI scripts, cookies, Choice of Webservers, SOAP and web services

RubyTk – Simple Tk Application, widgets, Binding events, Canvas, scrolling

UNIT - II

Extending Ruby: Ruby Objects in C, the Jukebox extension, Memory allocation, Ruby Type System,Embedding Ruby to Other Languages, Embedding a Ruby Interpreter

UNIT - III

Introduction to PERL and Scripting:

Scripts and Programs, Origin of Scripting, Scripting Today, Characteristics of Scripting Languages, Uses for Scripting Languages, Web Scripting, and the universe of Scripting Languages. PERL- Names and Values, Variables, Scalar Expressions, Control Structures, arrays, list, hashes, strings, pattern andregular expressions, subroutines.

UNIT - IV

Advanced PERL:

Finer points of looping, pack and unpack, filesystem, eval, data structures, packages, modules, objects, interfacing to the operating system, Creating Internet ware applications, Dirty Hands Internet Programming, security Issues.

UNIT - V

TCL:

TCL Structure, syntax, Variables and Data in TCL, Control Flow, Data Structures, input/output, procedures, strings, patterns, files, Advance TCL- eval, source, exec and uplevel commands, Name spaces, trapping errors, event driven programs, making applications internet aware, Nuts and Bolts Internet Programming, Security Issues, C Interface.

Tk:

Tk-Visual Tool Kits, Fundamental Concepts of Tk, Tk by example, Events and Binding, Perl-Tk.

TEXT BOOKS:

1. The World of Scripting Languages, David Barron, Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O'Reilly
3. "Programming Ruby" The Pramatic Progammers guide by Dabve Thomas Second edition

REFERENCE BOOKS:

1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Leeand B. Ware (Addison Wesley) Pearson Education.
2. Perl by Example, E. Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O'Reilly, SPD.
4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
5. Perl Power, J. P. Flynt, Cengage Learning.

IT634PE: MOBILE APPLICATION DEVELOPMENT (Professional Elective – III)

B.Tech. III Year II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Prerequisites

1. Acquaintance with JAVA programming
2. A Course on DBMS

Course Objectives

- To demonstrate their understanding of the fundamentals of Android operating systems
- To improves their skills of using Android software development tools
- To demonstrate their ability to develop software with reasonable complexity on mobileplatform
- To demonstrate their ability to deploy software to mobile devices
- To demonstrate their ability to debug programs running on mobile devices

Course Outcomes

- Understand the working of Android OS Practically.
- Develop Android user interfaces
- Develop, deploy and maintain the Android Applications.
- Students will develop expertise in designing user interfaces using various layouts, including linear, relative, grid, and table layouts.
- Understand and implement broadcast receivers, using intent filters to service implicit intents and handling notifications.
- Effectively create tables, insert, retrieve, and update data, and work with content providers for data management.

UNIT - I

Introduction to Android Operating System: Android OS design and Features – Android development framework, SDK features, Installing and running applications on Android Studio, Creating AVDs, Typesof Android applications, Best practices in Android programming, Android tools Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus etc, Resources for different devices and languages, Runtime Configuration Changes

Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes.

UNIT - II

Android User Interface: Measurements – Device and pixel density independent measuring unit - sLayouts – Linear, Relative, Grid and Table Layouts

User Interface (UI) Components –Editable and non-editable TextViews, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers

Event Handling – Handling clicks or changes of various UI components

Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities.

UNIT - III

Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS.

Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding andusing Intents received within an Activity

Notifications – Creating and Displaying notifications, Displaying Toasts.

UNIT - IV

Persistent Storage: Files – Using application specific folders and files, creating files, reading data from files, listing contents of a directory Shared Preferences – Creating shared preferences, saving andretrieving data using Shared Preference.

UNIT - V

Database: Introduction to SQLite database, creating and opening a database, creating tables, inserting retrieving and etindelg data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update).

TEXT BOOK:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012

REFERENCE BOOKS:

1. Android Application Development for Java Programmers, James C Sheusi, Cengage Learning, 2013
2. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013

IT635PE: SOFTWARE TESTING METHODOLOGIES (Professional Elective – III)

B.Tech. III Year II Sem.	**L T P C**
	3 0 0 3

Prerequisites

1. Software Engineering

Course Objectives

- To provide knowledge of the concepts in software testing such as testing process, criteria,strategies, and methodologies.
- To develop skills in software test automation and management using the latest tools.

Course Outcomes:

- Understand purpose of testing and path testing.
- Understand strategies in data flow testing and domain testing.
- Develop logic-based test strategies.
- Understand graph matrices and its applications.
- Implement test cases using any testing automation tool.
- Verify and validate both internal & external data input to the system using syntax testing.

UNIT - I

Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II

Transaction Flow Testing: Transaction flows, transaction flow testing techniques.
Data Flow testing: Basics of data flow testing, strategies in data flow testing, application of data flowtesting.

UNIT - III

Domain Testing: Domains and paths, Nice & ugly domains, domain testing, domains and interfacestesting, domain and interface testing, domains and testability.

Paths, Path products and Regular expressions: path products & path expression, reduction procedure,applications, regular expressions & flow anomaly detection.

UNIT – IV

Syntax Texting: Why, What How, A Grammar for formats, Test Case Generation, Implementation and Application and Testability Tips.
Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.
State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing,Testability tips.

UNIT - V

Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like Jmeter/selenium/soapUI/Catalon).

TEXT BOOKS:

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

REFERENCE BOOKS:

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

IT611OE: JAVA PROGRAMMING (Open Elective – I)

B.Tech. III Year II Sem.	L	T	P	C
	3	0	0	3

Course Objectives:

- To introduce object-oriented programming principles and apply them in solving problems.
- To introduce the implementation of packages and interfaces.
- To introduce the concepts of exception handling and multithreading.
- To introduce the design of Graphical User Interface using swing controls.

Course Outcomes:

- Able to solve real world problems using OOP techniques.
- Able to solve problems using java collection framework and I/O classes.
- Develop skills in creating and utilizing packages in Java
- Able to develop multithreaded applications with synchronization.
- Exhibit competence in applying Java Database Connectivity (JDBC) concepts
- Able to design GUI based applications.

UNIT - I

Foundations of Java: History of Java, Java Features, Variables, Data Types, Operators, Expressions, Control Statements. Elements of Java - Class, Object, Methods, Constructors and Access Modifiers, Generics, Inner classes, String class and Annotations.

OOP Principles: Encapsulation - concept, setter and getter method usage, this keyword. Inheritance - concept, Inheritance Types, super keyword. Polymorphism - concept, Method Overriding usage and Type Casting. Abstraction – concept, abstract keyword and Interface.

UNIT – II

Exception Handling: Exception and Error, Exception Types, Exception Handler, Exception Handling Clauses – try, catch, finally, throws and the throw statement, Built-in-Exceptions and Custom Exceptions.

Files and I/O Streams: The file class, Streams, The Byte Streams, Filtered Byte Streams, The Random Access File class.

UNIT – III

Packages- Defining a Package, CLASSPATH, Access Specifiers, importing packages. Few Utility Classes - String Tokenizer, BitSet, Date, Calendar, Random, Formatter, Scanner.

Collections: Collections overview, Collection Interfaces, Collections Implementation Classes, Sortingin Collections, Comparable and Comparator Interfaces.

UNIT – IV

Multithreading: Process and Thread, Differences between thread-based multitasking and process- based multitasking, Java thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication.

Java Database Connectivity: Types of Drivers, JDBC architecture, JDBC Classes and Interfaces, Basic steps in Developing JDBC Application, Creating a New Database and Table with JDBC.

UNIT - V

GUI Programming with Swing – Introduction, limitations of AWT, MVC architecture, components, containers, Layout Manager Classes, Simple Applications using AWT and Swing.

Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes,Handling mouse and keyboard events, Adapter classes.

TEXT BOOKS:

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

REFERENCE BOOKS:

1. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley & sons
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
3. Object Oriented Programming through Java, P. Radha Krishna, University Press.
4. Programming in Java, S. Malhotra, S. Chudhary, 2nd edition, Oxford Univ. Press.
5. Java Programming and Object-oriented Application Development, R. A. Johnson, Cengage Learning.

IT612OE: OBJECT ORIENTED PROGRAMMING USING C++ (Open Elective – I)

B.Tech. III Year II Sem.	**L**	**T**	**P**	**C**
	3	**0**	**0**	**3**

Course Objectives:

- Introduces Object Oriented Programming concepts using the C++ language.
- Understand the principles of data abstraction, inheritance and polymorphism;
- Implementation of the principles of virtual functions and polymorphism
- Handling formatted I/O, unformatted I/O in C++ and implementation of exception handling

Course Outcomes:

- Develop programs with reusability and understand OO functions.
- Develop programs for file handling, data abstraction, data hiding.
- Develop inheritance, overloading and exceptions in programming.
- Implement I/O operations and file handling.
- Develop applications for a range of problems using object-oriented programming techniques.
- Develop Robust Error Management in Programs.

UNIT - I

Object-Oriented Thinking: Different paradigms for problem solving, need for OOP paradigm, differences between OOP and Procedure oriented programming, Overview of OOP concepts Abstraction, Encapsulation, Inheritance and Polymorphism.

C++ Basics: Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures, References. Flow control statement- if, switch, while, for, do, break, continue, goto statements. Functions - Scope of variables, Parameter passing, Default arguments, inline functions, Recursive functions, Pointers to functions. Dynamic memory allocation and deallocation operators-new and delete, Preprocessor directives.

UNIT - II

C++ Classes and Data Abstraction: Class definition, Class structure, Class objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Dynamic creation and destruction of objects, Data abstraction, ADT and information hiding.

UNIT - III

Inheritance: Defining a class hierarchy, Different forms of inheritance, Defining the Base and Derived classes, Access to the base class members, Base and Derived class construction, Destructors, Virtualbase class.

Virtual Functions and Polymorphism: Static and Dynamic binding, virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes, Implications of polymorphic use of classes, Virtual destructors.

UNIT - IV

C++ I/O: I/O using C functions, Stream classes hierarchy, Stream I/O, File streams and String streams, Overloading operators, Error handling during file operations, Formatted I/O.

UNIT - V

Exception Handling: Benefits of exception handling, Throwing an exception, The try block, Catching an exception, Exception objects, Exception specifications, Stack unwinding, Rethrowing an exception, Catching all exceptions.

TEXT BOOKS:

1. The Complete Reference C++, 4th Edition, Herbert Schildt, Tata McGraw Hill.
2. Problem solving with C++: The Object of Programming, 4th Edition, Walter Savitch, PearsonEducation.

REFERENCE BOOKS:

1. The C++ Programming Language, 3rd Edition, B. Stroustrup, Pearson Education.
2. OOP in C++, 3rd Edition, T. Gaddis, J. Walters and G. Muganda, Wiley DreamTech Press.
3. Object Oriented Programming in C++, 3rd Edition, R. Lafore, Galgotia Publications Pvt Ltd.

IT604PC: COMPILER DESIGN LAB

B.Tech. III Year II Sem.

L	T	P	C
0	0	2	1

Prerequisites

1. A Course on "Object Oriented Programming through Java / C Programming".

Course Objectives:

- To understand the various phases in the design of a compiler.
- To understand the design of top-down and bottom-up parsers.
- To understand syntax directed translation schemes.
- To introduce lex and yacc tools.

Course Outcomes:

- Design, develop, and implement a compiler for any language.
- Use lex and yacc tools for developing a scanner and a parser.
- Design and implement LL and LR parsers.

List of Experiments:

1. Implementation of symbol table.
2. Develop a lexical analyzer to recognize a few patterns inc (ex. Identifiers, constants, comments, operators etc.)
3. Implementation of lexical analyzer using Lex tool.
4. Generate Yacc specification for a few syntactic categories.
   a) Program to recognize a valid arithmetic expression that uses operator +,-, * and /.
   b)Program to recognize a valid variable which starts with a letter followed by any numberof letter or digits.
   c) Implementation of calculator using Lex and Yacc.
5. Convert the BNF rules into Yacc form and write code to generate abstract syntax tree.
6. Implement Type checking.
7. Implement any one storage allocation strategies (heap, stack, static)
8. Write a Lex program to count the number of words and number of lines in a given file or program.
9. Write a 'C' program to implement lexical analyzer using c program.
10. Write recursive descent parser for the grammar
    E->E+T E->T T->T*F T->FF->(E)|id.
11. Write recursive descent parser for the grammar S->(L) S->a L->L,S L->S
12. Write a C program to calculate First function for the grammar
    E->E+T E->T   T->T*F T->F   F->(E)|id
13. Write a YACC program to implement a top down parser for the given grammar: S->AA A->aA|a.
14. Write a YACC program to evaluate algebraic expression.

TEXT BOOK:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam,Ravi Sethi, Jeffry D. Ullman.

REFERENCE BOOKS:

1. Lex & Yacc – John R. Levine, Tony Mason, Doug Brown, O'reilly
2. Compiler Construction, Louden, Thomson.

IT605PC: EMBEDDED SYSTEMS LAB

B.Tech. III Year II Sem.

L	T	P	C
0	0	2	1

Pre-requisites:

1. A course on “Digital Logic Design and Microprocessors”.
2. A course on “Computer Organization and Architecture”.

Course Objectives:

- To provide an overview of principles of Embedded System
- To provide a clear understanding of role of firmware, operating systems in correlation withhardware systems.

Course Outcomes:

- Expected to understand the selection procedure of processors in the embedded domain.
- Design procedure of embedded firm ware.
- Expected to visualize the role of real-time operating systems in embedded systems.
- Expected to evaluate the correlation between task synchronization and latency issues

The following experiments have to be executed using
i) Microcontroller Kits (8051/ Raspberry Pi /Arduino)
(ii) Use embedded C/ Python/ assembly language.

List of Experiments:

1. Programs to perform arithmetic, logical, branching, and loop operations by a microcontroller.
2. Generate time delay using timers in a microcontroller.
3. Write a C program to count the number of times the switch is pressed and released.
4. Illustrate the use of a port header file (Port M) using an interface consisting of a keypad and 7- Segment Display.
5. write a program to display "Hello world” on display of the receiving microcontroller using RS232.
6. Flash the operating System onto the device into a stable functional state by porting desktopenvironment with necessary packages.
7. Program available GPIO Plus of the corresponding device using native programming language,interface LEDs and interface LED / Switches and test it's functionality.
8. Using the light sensor, monitor the light intensity and automatically turn ON/OFF LED.(forPredefined threshold light intensity value)
9. Dice game simulation- generate a random value Similar to dice value and display the sameusing 16x2 LCD.
10. Export display to other system using available desktop display as display for the device usingSSH client and X11 display seven.
11. Hosting a website on Board- Build and host a simple website (static) on the device and make itaccessible online. (Need to install Sewn (e.g., Apache).)
12. Interface a regular USB webcam to the device and turn it into fully functional IP

webcam Note: Devices include Arduino, Raspberry Pi, and BeagleBour.

TEXT BOOK:

1. Embedded Systems, Raj Kamal, 2nd edition, Tata Mc Graw Hill
2. Shibu K V, “Introduction to Embedded Systems”, Second Edition, Mc Graw Hill

REFERENCE BOOKS:

1. Rajkamal, Embedded Systems Architecture, Programming and Design, Tata McGraw-Hill
2. Frank Vahid and Tony Givargis, “Embedded Systems Design” - A Unified Hardware/SoftwareIntroduction, John Wiley
3. Lyla, “Embedded Systems” -Pearson
4. David E. Simon, An Embedded Software Primer, Pearson Education Asia, First Indian Reprint2000.

IT611PE: FULL STACK DEVELOPMENT LAB (Professional Elective – III)

B.Tech. III Year II Sem.

L	T	P	C
0	0	2	1

Pre-Requisites:

1. Object Oriented Programming
2. Web Technologies

Course Objectives:

- Introduce fast, efficient, interactive and scalable web applications using run time environmentprovided by the full stack components.

Course Outcomes:

- Design flexible and responsive Web applications using Node JS, React, Express and Angular.
- Perform CRUD operations with MongoDB on huge amount of data.
- Develop real time applications using react components.
- Use various full stack modules to handle http requests and responses.

List of Experiments:

1. Create an application to setup node JS environment and display “Hello World”.
2. Create a Node JS application for user login system.
3. Write a Node JS program to perform read, write and other operations on a file.
4. Write a Node JS program to read form data from query string and generate response usingNodeJS
5. Create a food delivery website where users can order food from a particular restaurant listed inthe website for handling http requests and responses using NodeJS.
6. Implement a program with basic commands on databases and collections using MongoDB.
7. Implement CRUD operations on the given dataset using MongoDB.
8. Perform Count, Limit, Sort, and Skip operations on the given collections using MongoDB.
9. Develop an angular JS form to apply CSS and Events.
10. Develop a Job Registration form and validate it using angular JS.
11. Write an angular JS application to access JSON file data of an employee from a server using $http service.
12. Develop a web application to manage student information using Express and Angular JS.
13. Write a program to create a simple calculator Application using React JS.
14. Write a program to create a voting application using React JS
15. Develop a leave management system for an organization where users can apply different typesof leaves such as casual leave and medical leave. They also can view the available number ofdays using react application.
16. Build a music store application using react components and provide routing among the web pages.
17. Create a react application for an online store which consist of registration, login, product information pages and implement routing to navigate through these pages.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas., React in Action, 1st Edition, Manning Publications.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo,Express, React, and Node, 2nd Edition, Apress, 2019.
2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday SkillsExpected of a Modern Full Stack Web Developer’, 1st edition, Apress, 2018.
3. Brad Green& Seshadri. Angular JS. 1st Edition. O'Reilly Media, 2013.
4. Kirupa Chinnathambi, Learning React: A Hands-On Guide to Building Web Applications UsingReact and Redux, 2nd edition, Addison-Wesley Professional, 2018.

IT612PE: DATA MINING LAB (Professional Elective – III)

B.Tech. III Year II Sem. **L T P C**
0 0 2 1

Prerequisites

- A course on "Database Management System".

Course Objectives:

- The course is intended to obtain hands-on experience using data mining software.
- Intended to provide practical exposure of the concepts in data mining algorithms.

Course Outcomes:

- Apply preprocessing statistical methods for any given raw data.
- Gain practical experience of constructing a data warehouse.
- Implement various algorithms for data mining in order to discover interesting patternsfrom large amounts of data.
- Apply OLAP operations on data cube construction.

LIST OF EXPERIMENTS: Experiments using Weka/ Pentaho/Python

1. Data Processing Techniques:
   (i) Data cleaning (ii) Data transformation – Normalization (iii) Data integration
2. Partitioning - Horizontal, Vertical, Round Robin, Hash based.
3. Data Warehouse schemas – star, snowflake, fact constellation.
4. Data cube construction – OLAP operations.
5. Data Extraction, Transformations & Loading operations.
6. Implementation of Attribute oriented induction algorithm.
7. Implementation of apriori algorithm.
8. Implementation of FP – Growth algorithm.
9. Implementation of Decision Tree Induction.
10. Calculating Information gain measures.
11. Classification of data using Bayesian approach.
12. Classification of data using K – nearest neighbour approach.
13. Implementation of K – means algorithm.
14. Implementation of BIRCH algorithm.
15. Implementation of PAM algorithm.
16. Implementation of DBSCAN algorithm.

TEXT BOOKS:

1. Data Mining – Concepts and Techniques - JIAWEI HAN &MICHELINE KAMBER, Elsevier.
2. Data Warehousing, Data Mining &OLAP- Alex Berson and Stephen J. Smith- Tata McGraw-HillEdition, Tenth reprint 2007

REFERENCE BOOK:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Anuj Karpatne, Introduction to Data Mining,Pearson Education

IT613PE: SCRIPTING LANGUAGES LAB (Professional Elective – III)

B.Tech. III Year II Sem.

L	T	P	C
0	0	2	1

Prerequisites: Any High level programming language (C, C++).

Course Objectives

- To Understand the concepts of scripting languages for developing web based projects.
- To understand the applications the of Ruby, TCL, Perl scripting languages.

..

Course Outcomes

- Ability to understand the differences between Scripting languages and programming languages.
- Gain some fluency programming in Ruby, Perl, TCL.

LIST OF EXPERIMENTS

1.Write a Ruby script to create a new string which is n copies of a given string where n is a non-negative integer.

2. Write a Ruby script which accept the radius of a circle from the user and compute the parameter and area.

3. Write a Ruby script which accept the users first and last name and print them in reverse order with aspace between them.

4. Write a Ruby script to accept a filename from the user print the extension of that

5. Write a Ruby script to find the greatest of three numbers.

6. Write a Ruby script to print odd numbers from 10 to 1.

7. Write a Ruby script to check two integers and return true if one of them is 20 otherwise return theirsum.

8. Write a Ruby script to check two temperatures and return true if one is less than 0 and the other isgreater than 100.

9. Write a Ruby script to print the elements of a given array.

10. Write a Ruby program to retrieve the total marks where subject name and marks of a student storedin a hash.

11. Write a TCL script to find the factorial of a number.

12. Write a TCL script that multiplies the numbers from 1 to 10.

13. Write a TCL script for sorting a list using a comparison function.

14. Write a TCL script to (i) create a list (ii) append elements to the list (iii) Traverse the list (iv)Concatenate the list.

15. Write a TCL script to comparing the file modified times.

16. Write a TCL script to Copy a file and translate to native format.

17. a) Write a Perl script to find the largest number among three numbers.
    b) Write a Perl script to print the multiplication tables from 1-10 using subroutines.

18. Write a Perl program to implement the following list of manipulating functions
    a) Shift
    b) Unshift
    c) Push

19. a) Write a Perl script to substitute a word, with another word in a string.
    b) Write a Perl script to validate IP address and email address.

20. Write a Perl script to print the file in reverse order using command line arguments

TEXT BOOKS:

1. The World of Scripting Languages, David Barron,Wiley Publications.
2. Ruby Programming language by David Flanagan and Yukihiro Matsumoto O’Reilly
3. “Programming Ruby” The Pramatic Progammers guide by Dabve Thomas Second edition.

REFERENCE BOOKS:

1. Open Source Web Development with LAMP using Linux Apache, MySQL, Perl and PHP, J.Leeand B. Ware (Addison Wesley) Pearson Education.
2. Perl by Example, E. Quigley, Pearson Education.
3. Programming Perl, Larry Wall, T. Christiansen and J. Orwant, O’Reilly, SPD.
4. Tcl and the Tk Tool kit, Ousterhout, Pearson Education.
5. Perl Power, J. P. Flynt, Cengage Learning.

IT614PE: MOBILE APPLICATION DEVELOPMENT LAB (Professional Elective – III)

B.Tech. III Year II Sem. **L T P C**
0 0 2 1

Prerequisites: --- NIL---

Course Objectives:

- To learn how to develop Applications in an android environment.
- To learn how to develop user interface applications.
- To learn how to develop URL related applications.

Course Outcomes:

- Understand the working of Android OS Practically.
- Develop user interfaces.
- Develop, deploy and maintain the Android Applications.

LIST OF EXPERIMENTS:

1. (a) Create an Android application that shows Hello + name of the user and run it on an emulator.
   (b) Create an application that takes the name from a text box and shows hello message along with thename entered in the text box, when the user clicks the OK button.
2. Create a screen that has input boxes for User Name, Password, Address, Gender (radio buttons for male and female), Age (numeric), Date of Birth (Datepicker), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button. Use (a) Linear Layout (b) Relative Layout and (c) Grid Layout or Table Layout.
3. Develop an application that shows names as a list and on selecting a name it should show the details of the candidate on the next screen with a “Back” button. If the screen is rotated to landscape mode (width greater than height), then the screen should show list on left fragment and details on the right fragment instead of the second screen with the back button. Use Fragment transactions and Rotation event listeners.
4. Develop an application that uses a menu with 3 options for dialing a number, opening a website and to send an SMS. On selecting an option, the appropriate action should be invoked using intents.
5. Develop an application that inserts some notifications into Notification area and whenever a notification is inserted, it should show a toast with details of the notification.
6. Create an application that uses a text file to store usernames and passwords (tab separated fields and one record per line). When the user submits a login name and password through a screen, thedetails should be verified with the text file data and if they match, show a dialog saying that login issuccessful. Otherwise, show the dialog with a Login Failed message.
7. Create a user registration application that stores the user details in a database table.
8. Create a database and a user table where the details of login names and passwords are stored. Insert some names and passwords initially. Now the login details entered by the user should be verified with the database and an appropriate dialog should be shown to the user.
9. Create an admin application for the user table, which shows all records as a list and the admin canselect any record for edit or modify. The results should be reflected in the table.
10. Develop an application that shows all contacts of the phone along with details like name, phone number, mobile number etc.
11. Create an application that saves user information like name, age, gender etc. in

shared preferenceand retrieves them when the program restarts.

12. Create an alarm that rings every Sunday at 8:00 AM. Modify it to use a time picker to set alarm time.

TEXT BOOKS:

1. Professional Android 4 Application Development, Reto Meier, Wiley India, (Wrox), 2012.
2. Android Application Development for Java Programmers, James C Sheusi, Cengage, 2013.

REFERENCE BOOK:

1. Beginning Android 4 Application Development, Wei-Meng Lee, Wiley India (Wrox), 2013.

IT615PE: SOFTWARE TESTING METHODOLOGIES LAB (Professional Elective – III)

B.Tech. III Year II Sem.	**L T P C**
	0 0 2 1

Prerequisites

- A basic knowledge of programming.

Course Objectives

- To provide knowledge of software testing methods.
- To develop skills in automation of software testing and software test automation managementusing the latest tools.

Course Outcomes

- Design and develop the best test strategies in accordance with the development model.
- Design and develop GUI, Bitmap and database checkpoints.
- Develop database checkpoints for different checks.
- Perform batch testing with and without parameter passing.

List of Experiments

1. Recording in context sensitive mode and analog mode.
2. GUI checkpoint for single property.
3. GUI checkpoint for single object/window.
4. GUI checkpoint for multiple objects.
5.
   a. Bitmap checkpoint for object/window.
   b. Bitmap checkpoint for screen area.
6. Database checkpoint for Default check.
7. Database checkpoint for custom check.
8. Database checkpoint for runtime record check.
9.
   a. Data driven test for dynamic test data submission.
   b. Data driven test through flat files.
   c. Data driven test through front grids.
   d. Data driven test through excel test.
10.
   a. Batch testing without parameter passing.
   b. Batch testing with parameter passing.
11. Data driven batch.
12. Silent mode test execution without any interruption.
13. Test case for calculator in windows application.

TEXT BOOKS

1. Software Testing techniques, Baris Beizer, 2nd Edition, Dreamtech.
2. Software Testing Tools, Dr. K.V.K.K.Prasad, Dreamtech.

REFERENCE BOOKS

1. The craft of software testing, Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille).
3. Software Testing in the Real World, Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing, Meyers, John Wiley.

IT606PC: BIG DATA-SPARK

B.Tech. III Year II Sem. **L T P C**
0 0 4 2

Course Objectives:

- The main objective of the course is to process Big Data with advance. architecture like sparkand streaming data in Spark.

Course Outcomes:

0. Develop MapReduce Programs to analyze large dataset Using Hadoop and Spark.
1. Write Hive queries to analyze large dataset Outline the Spark Ecosystem and its components.
2. Perform the filter, count, distinct, map, flatMap RDD Operations in Spark.
3. Build Queries using Spark SQL.
4. Apply Spark joins on Sample Data Sets.
5. Make use of sqoop to import and export data from hadoop to database and vice-versa.

List of Experiments:

1. To Study of Big Data Analytics and Hadoop Architecture.
   (i) know the concept of big data architecture.
   (ii) know the concept of Hadoop architecture.

2. Loading DataSet in to HDFS for Spark Analysis
   Installation of Hadoop and cluster management
   (i) Installing Hadoop single node cluster in ubuntu environment
   (ii) Knowing the differencing between single node clusters and multi-node clusters
   (iii) Accessing WEB-UI and the port number
   (iv) Installing and accessing the environments such as hive and sqoop

3. File management tasks & Basic linux commands
   (i) Creating a directory in HDFS
   (ii) Moving forth and back to directories
   (iii) Listing directory contents
   (iv) Uploading and downloading a file in HDFS
   (v) Checking the contents of the file
   (vi) Copying and moving files
   (vii) Copying and moving files between local to HDFS environment
   (viii) Removing files and paths
   (ix) Displaying few lines of a file
   (x) Display the aggregate length of a file
   (xi) Checking the permissions of a file
   (xii) Zipping and unzipping the files with & without permission pasting it to a location
   (xiii) Copy, Paste commands

4. Map-reducing
   (i) Definition of Map-reduce
   (ii) Its stages and terminologies
   (iii) Word-count program to understand map-reduce (Mapper phase, Reducer phase, Drivercode)
5. Implementing Matrix-Multiplication with Hadoop Map-reduce

6. Compute Average Salary and Total Salary by Gender for an Enterprise.

7. (i) Creating hive tables (External and internal)
   (ii) Loading data to external hive tables from sql tables(or)Structured c.s.v using scoop
   (iii) Performing operations like filterations and updations

(iv) Performing Join (inner, outer etc)
(v) Writing User defined function on hive tables

8. Create a sql table of employees Employee table with id,designation Salary table (salary ,dept id) Create external table in hive with similar schema of above tables,Move data to hive using scoop and load the contents into tables,filter a new table and write a UDF to encrypt the table with AES-algorithm, Decrypt it with key to show contents

9. (i) Pyspark Definition(Apache Pyspark) and difference between Pyspark, Scala, pandas
(ii) Pyspark files and class methods
(iii) get(file name)
(iv) get root directory()

10. Pyspark -RDD'S
(i) what is RDD's?
(ii) ways to Create RDD
(iii) parallelized collections
(iv) external dataset
(v) existing RDD's
(vi) Spark RDD's operations (Count, foreach(), Collect, join,Cache()

11. Perform pyspark transformations
(i) map and flatMap
(ii) to remove the words, which are not necessary to analyze this text.
(iii) groupBy
(iv) What if we want to calculate how many times each word is coming in corpus ?
(v) How do I perform a task (say count the words 'spark' and 'apache' in rdd3) separatly oneach partition and get the output of the task performed in these partition ?
(vi) unions of RDD
(vii) join two pairs of RDD Based upon their key

12. Pyspark sparkconf-Attributes and applications
(i) What is Pyspark spark conf ()
(ii) Using spark conf create a spark session to write a dataframe to read details in a c.s.v andlater move that c.s.v to another location

TEXT BOOKS:

1. Spark in Action, Marko Bonaci and Petar Zecevic, Manning.
2. PySpark SQL Recipes: With HiveQL, Dataframe and Graphframes, Raju Kumar Mishra and Sundar Rajan Raman, Apress Media.

WEB LINKS:

1. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0133015058445 18912251 8 2_shared/overview
2. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01258388119638 835242_s hared/overview
3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_0126052684230 08256169 2 _shared/overview

*MC609: ENVIRONMENTAL SCIENCE

B.Tech. III Year II Sem.

L	T	P	C
3	0	0	0

Course Objectives:

- Understanding the importance of ecological balance for sustainable development.
- Understanding the impacts of developmental activities and mitigation measures
- Understanding the environmental policies and regulations

Course Outcomes: Based on this course, the Engineering graduate will understand /evaluate / develop technologies on the basis of ecological principles and environmental regulations which in turn helps insustainable development

UNIT - I

Ecosystems: Definition, Scope and Importance of ecosystem. Classification, structure, and function ofan ecosystem, Food chains, food webs, and ecological pyramids. Flow of energy, Biogeochemical cycles, Bioaccumulation, Biomagnification, ecosystem value, services and carrying capacity, Field visits.

UNIT - II

Natural Resources: Classification of Resources: Living and Non-Living resources, **water resources:** use and over utilization of surface and ground water, floods and droughts, Dams: benefits and problems. **Mineral resources:** use and exploitation, environmental effects of extracting and using mineral resources, **Land resources:** Forest resources, **Energy resources:** growing energy needs, renewable and non renewable energy sources, use of alternate energy source, case studies.

UNIT - III

Biodiversity And Biotic Resources: Introduction, Definition, genetic, species and ecosystem diversity. Value of biodiversity; consumptive use, productive use, social, ethical, aesthetic and optional values. India as a mega diversity nation, Hot spots of biodiversity. Field visit. Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts; conservation of biodiversity: In-Situ and Ex-situ conservation. National Biodiversity act.

UNIT - IV

Environmental Pollution and Control Technologies: Environmental Pollution: Classification of pollution, **Air Pollution:** Primary and secondary pollutants, Automobile and Industrial pollution, Ambient air quality standards. **Water pollution:** Sources and types of pollution, drinking water quality standards. **Soil Pollution:** Sources and types, Impacts of modern agriculture, degradation of soil. **Noise Pollution:** Sources and Health hazards, standards, **Solid waste:** Municipal Solid Waste management, composition and characteristics of e-Waste and its management. **Pollution control technologies:** Wastewater Treatment methods: Primary, secondary and Tertiary.

Overview of air pollution control technologies, Concepts of bioremediation. **Global Environmental Problems and Global Efforts: C**limate change and impacts on human environment. Ozone depletion and Ozone depleting substances (ODS). Deforestation and desertification. International conventions / Protocols: Earth summit, Kyoto protocol, and Montréal Protocol.

UNIT - V

Environmental Policy, Legislation & EIA: Environmental Protection act, Legal aspects Air Act- 1981, Water Act, Forest Act, Wild life Act, Municipal solid waste management and handling rules, biomedicalwaste management and handling rules, hazardous waste management and handling rules. EIA: EIA structure, methods of baseline data acquisition. Overview on Impacts of air, water, biological and Socio- economical aspects. Strategies for risk assessment, Concepts of Environmental Management Plan

(EMP). **Towards Sustainable Future:** Concept of Sustainable Development, Population and its explosion, Crazy Consumerism, Environmental Education, Urban Sprawl, Human health, Environmental Ethics, Concept of Green Building, Ecological Foot Print, Life Cycle assessment (LCA),Low carbon life style.

TEXT BOOKS:

1. Textbook of Environmental Studies for Undergraduate Courses by Erach Bharucha forUniversity Grants Commission.
2. Environmental Studies by R. Rajagopalan, Oxford University Press.

REFERENCE BOOKS:

1. Environmental Science: towards a sustainable future by Richard T. Wright. 2008 PHL LearningPrivate Ltd. New Delhi.
2. Environmental Engineering and science by Gilbert M. Masters and Wendell P. Ela. 2008 PHILearning Pvt. Ltd.
3. Environmental Science by Daniel B. Botkin & Edward A. Keller, Wiley INDIA edition.
4. Environmental Studies by Anubha Kaushik, 4th Edition, New age international publishers.
5. Text book of Environmental Science and Technology - Dr. M. Anji Reddy 2007, BS Publications.